# Experiment 5: Transformations and Motions

CZ2003 Computer Graphics and Visualization

**SS3**



| Name | Matric Number |
|---|---|
| Pang Yu Shao | U1721680D |

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE


6th April 2021

## Contents

# 1 TRANSFORMING A DEFINED 3-D CURVE

## 1.1 Deriving Transformation Matrix

First, a transformation matrix is defined for rotating by $\frac{\pi}{2}$ about an axis parallel to axis Y, and passing through the point with coordinates $(M + 5, 0, 0)$. Since M = 10, $(15, 0, 0)$.

Therefore to achieve this, first we align the axis with the Y-axis by translation of $(-15, 0, 0)$. Then, rotation of $\frac{\pi}{2}$ radians about axis Y is applied. Lastly, the initial translation is undone by applying a translation of $(15, 0, 0)$.

The three transformations can be defined using the following matrices:

$$T_1 = \begin{bmatrix} 1 & 0 & 0 & -15 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R = \begin{bmatrix} cos(\frac{\pi}{2}) & 0 & sin(\frac{\pi}{2}) & 0 \\ 0 & 1 & 0 & 0 \\ -sin(\frac{\pi}{2}) & 0 & cos(\frac{\pi}{2}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2 = \begin{bmatrix} 1 & 0 & 0 & 15 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The final transformation matrix, M can then be obtained:
$M = T_2 \cdot R \cdot T_1$

$$M = \begin{bmatrix} 1 & 0 & 0 & 15 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -15 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 15 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 15 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 1 & 15 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 15 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

## 1.2 Applying Transformation Matrix to Parametrically Defined Curve

The matrix obtained from section 1.1 is to be applied to the parametric equations of the curve defined in experiment 1, exercise 3. The following parametric functions defined the abovementioned curve.

$$\mathbf{x_0(u) = (8 - 15cos(2\pi u))cos(2\pi u)}$$
$$\mathbf{y_0(u) = (8 - 15cos(2\pi u))sin(2\pi u)}$$
$$\mathbf{z_0(u) = 0}, u \in [0, 1]$$

To obtain the transformed parametric equations, matrix multiplication can be used:

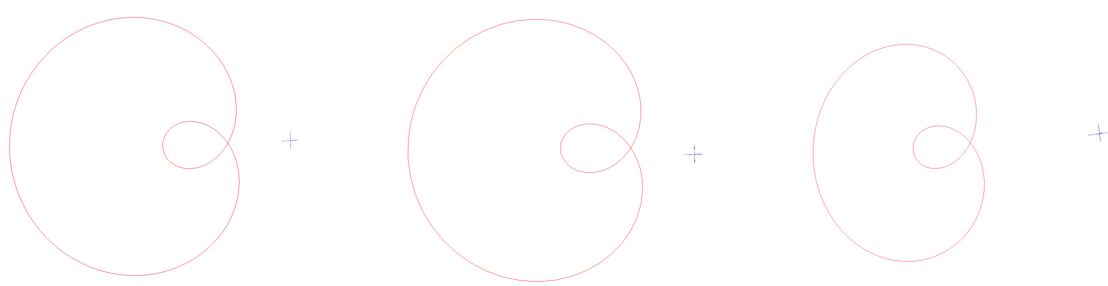
$$\begin{bmatrix} x(u) \\ y(u) \\ z(u) \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 15 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 15 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0(u) \\ y_0(u) \\ z_0(u) \\ 1 \end{bmatrix}$$

Therefore,

$$x(u) = z_0(u) + 15$$
$$y(u) = y_0(u)$$
$$z(u) = -x_0(u) + z_0(u) + 15$$

$$\mathbf{x(u) = 15}$$
$$\mathbf{y(u) = (8 - 15cos(2\pi u))sin(2\pi u)}$$
$$\mathbf{z(u) = 15 - (8 - 15cos(2\pi u))cos(2\pi u)}$$
$$u \in [0, 1]$$

By plotting the parametric equations defined above, we get the following images:



(a) Resolution: 150

(b) Resolution: 75

(c) Resolution: 300

Fig. 1. Plots of the parametrically defined curve in "**1b.wrl**" with differing resolutions

As seen in Fig. 1 above, a sampling resolution of **150** is the optimal resolution for rendering the curve. When the resolution is reduced to 75, the polyline interpolations can be seen when zoomed in. When the resolution is increased to 300, there is no visible difference in the rendered curve when compared to that rendered with resolution of 150.

## 2 ANIMATING TRANSFORMATION MOTION WITH DECELERATION

By modfying the parametric definitions to include the time parameter, $t$, The rotation can be animated with the following definitions:

$$x(u,t) = (1-\tau(t)) * x_0(u) + \tau(t) * x_1(u)$$

$$y(u,t) = (1-\tau(t)) * y_0(u) + \tau(t) * y_1(u)$$

$$z(u,t) = (1-\tau(t)) * z_0(u) + \tau(t) * z_1(u)$$

Since $y_0(u) = y_1(u)$,

$$x(u,t) = ((1-\tau(t))(8-15cos(2\pi u))cos(2\pi u)) + (15\tau(t))$$

$$y(u,t) = (8-15cos(2\pi u))sin(2\pi u)$$

$$z(u,t) = \tau(t)(15-(8-15cos(2\pi u))cos(2\pi u))$$

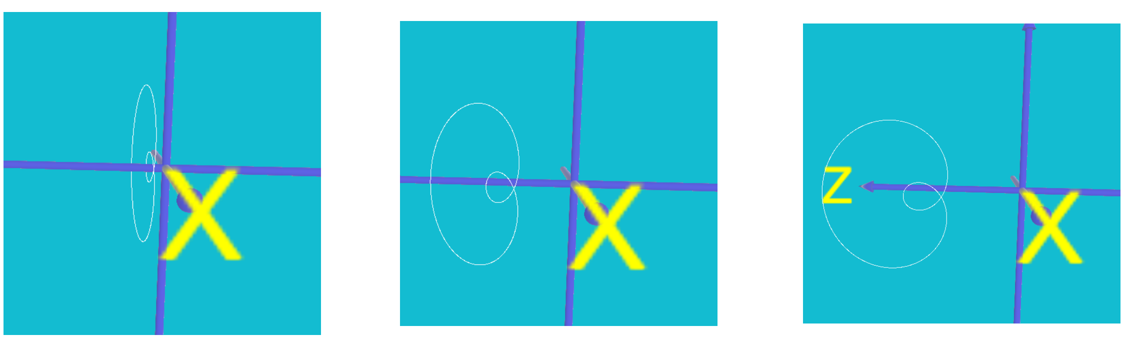
Lastly, the function to control the speed of animation (eceleration) is defined:

$$\tau(t) = sin(\frac{\pi}{2}\frac{t-t_1}{t_2-t_1})$$

Since $t_1 = 0$ and $t_2 = 1$,

$$\tau(t) = sin(\frac{\pi}{2}t)$$

The defined functions were then animated using ShapeExplorer, with the resolution set to **150** since the shape is unchanged. The CycleInterval parameter is also set to **5** seconds. The curve can be seen to rotate from the X-Y plane until it is parallel with the Y-Z plane. The figure below shows the animation of the rotation.




(a) Start (b) Middle (c) End

Fig. 2. Frames of the animation defined in "**2.Func**" at different stages

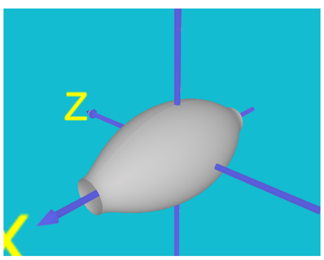
# 3 MORPHING BETWEEN TWO SURFACES

## 3.1 Converting and Displaying Surfaces

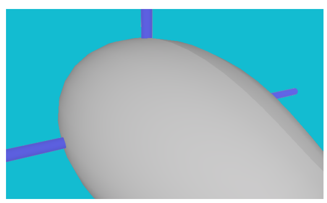
*3.1.1 Surface 10 (M).* From Table 3, Surface 10 is defined as:
$x = 0.5((\phi - 0.5sin(\phi)) - 3)$
$y = 0.5cos(4\alpha\pi)(1 - 0.5cos(\phi))$
$z = 0.5sin(4\alpha\pi)(1 - 0.5cos(\phi))$
$0 \leq \alpha \leq 0.5,\ 0 \leq \phi \leq 2\pi$

Therefore, they can be convered into the following parametric functions:
$x(u, v) = 0.5((2\pi u - 0.5sin(2\pi u)) - 3)$
$y(u, v) = 0.5cos(2\pi v)(1 - 0.5cos(2\pi u))$
$z(u, v) = 0.5sin(2\pi v)(1 - 0.5cos(2\pi u))$
$u, v \in [0, 1]$

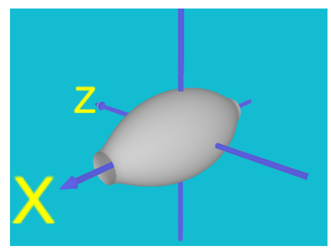
By entering the above functions onto ShapeExplorer, the following curves are rendered:




(a) Resolution: 50 50



(b) Resolution: 25 25




(c) Resolution: 100 100

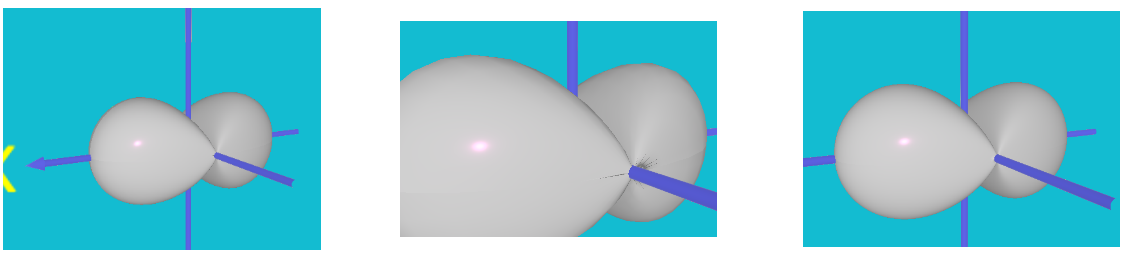
Fig. 3. Plots of the parametrically defined surface in "**3a1.Func**" with differing resolutions

As seen in Fig. 3 above, a sampling resolution of **50** for u and v is the optimal resolution for rendering the surface. When the resolution is reduced to 25, the surface is not smooth when zoomed in and the interpolations can be seen. When the resolution is increased to 100, there is no visible difference in the rendered surface when compared to that rendered with resolution of 50.

*3.1.2 Surface 18 (N+M).* From Table 3, Surface 18 is defined as:
$x = (cos(0.25\theta) + 1)cos(\phi)$
$y = sin(0.25\theta)cos(\phi)$
$z = sin(\phi)$
$0 \leq \theta \leq 8\pi,\ 0 \leq \phi \leq 2\pi$

Therefore, they can be convered into the following parametric functions:
$x(u, v) = (cos(2\pi u) + 1)cos(2\pi v)$
$y(u, v) = sin(2\pi u)cos(2\pi v)$
$z(u, v) = sin(2\pi v)$
$u, v \in [0, 1]$

By entering the above functions onto ShapeExplorer, the following curves are rendered:



(a) Resolution: 50 50

(b) Resolution: 25 25

(c) Resolution: 100 100

Fig. 4. Plots of the parametrically defined surface in "**3a2.Func**" with differing resolutions

As seen in Fig. 4 above, a sampling resolution of **50** for u and v is the optimal resolution for rendering the surface. When the resolution is reduced to 25, the surface is not smooth when zoomed in and the interpolations can be seen. When the resolution is increased to 100, there is no visible difference in the rendered surface when compared to that rendered with resolution of 50.

### 3.2 Morphing Between Two Defined Surfaces

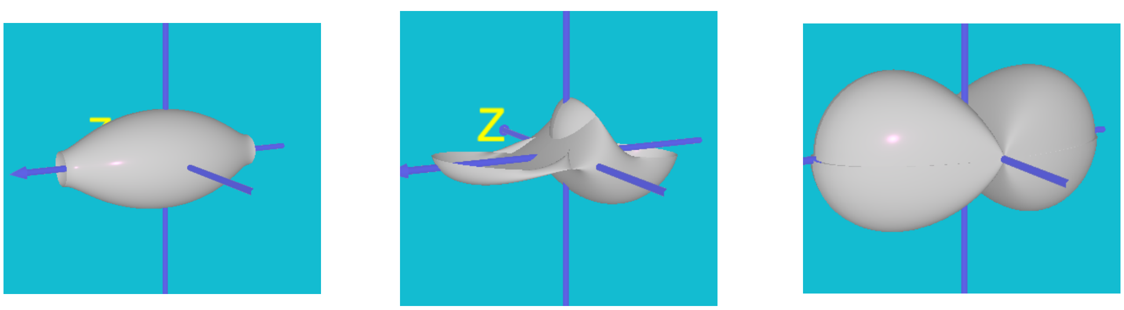
From the above section, we obtained the following functions:
**Surface 10:**
$x_0(u, v) = 0.5((2\pi u - 0.5sin(2\pi u)) - 3)$
$y_0(u, v) = 0.5cos(2\pi v)(1 - 0.5cos(2\pi u))$
$z_0(u, v) = 0.5sin(2\pi v)(1 - 0.5cos(2\pi u))$
$u, v \in [0, 1]$

**Surface 18:**
$x_1(u, v) = (cos(2\pi u) + 1)cos(2\pi v)$
$y_1(u, v) = sin(2\pi u)cos(2\pi v)$
$z_1(u, v) = sin(2\pi v)$
$u, v \in [0, 1]$

Since the morphing animation is back and forth, the speed control function can be defined parametrically by:
$\tau(t) = 1 - |1 - 2t|$

Therefore, the morphing can be defined parametrically as:
$x(u, v, t) = x_0(u, v) + (x_1(u, v) - x_0(u, v))\tau(t)$
$y(u, v, t) = y_0(u, v) + (y_1(u, v) - y_0(u, v))\tau(t)$
$z(u, v, t) = z_0(u, v) + (z_1(u, v) - z_0(u, v))\tau(t)$
$u, v \in [0, 1]$

The morphing animation is rendered using ShapeExplorer with a resolution of **50** for both u and v as it is the optimal resolution found for both surfaces in the experiments carried out in the previous sections. The CycleInterval parameter is also set to **5** seconds. The figure below shows the animation of the rotation.




(a) Surface 10 (b) In-between (c) Surface 18

Fig. 5. Frames of the animation defined in "**3b.Func**" at different stages

Once the surface is morphed from surface 10 to surface 18, it morphs back to surface 10 as the speed control is defined to be uniform with back and forth effect.